Visual collective behaviors on spherical robots

Diego Castro, Christophe Eloy, Franck Ruffier

Visual collective behaviors on spherical robots

Diego Castro[1,2],Christophe Eloy[2] and Franck Ruffier[1,*]

[1]Aix Marseille Université, CNRS, ISM, Marseille, 13288, France
[2]Aix Marseille Université, CNRS, Centrale Med, IRPHE, Marseille, 13013, France
[*]Author to whom any correspondence should be addressed.

E-mail: franck.ruffier@cnrs.fr

Submitted to: *Bioinspir. Biomim.*
8 October 2024

Abstract.

The implementation of collective motion, traditionally, disregard the limited sensing capabilities of an individual, to instead assuming an omniscient perception of the environment. This study implements a visual flocking model in a "robot-in-the-loop" approach to reproduce these behaviors with a flock composed of 10 independent spherical robots. The model achieves robotic collective motion by only using panoramic visual information of each robot, such as retinal position, optical size and optic flow of the neighboring robots. We introduce a virtual anchor to confine the collective robotic movements so to avoid wall interactions. For the first time, a simple visual robot-in-the-loop approach succeed in reproducing several collective motion phases, in particular, swarming, and milling. Another milestone achieved with by this model is bridging the gap between simulation and physical experiments by demonstrating nearly identical behaviors in both environments with the same visual model. To conclude, we show that our minimal visual collective motion model is sufficient to recreate most collective behaviors on a robot-in-the-loop system that is scalable, behaves as numerical simulations predict and is easily comparable to traditional models.

Keywords: Flocking, robot-in-the-loop, Collective motion, Milling, Swarming, Optic Flow, Visual information.

1. Introduction

Biological collective motion has inspired efforts to understand, replicate, and ultimately utilize on artificial agents. Nearly 40 years ago, Reynolds *et al.,*[1] proposed that a set of simple, individual-level rules could explain collective motion. These rules include (i) attraction, the desire to remain close to other individuals; (ii) alignment, the tendency to move in the same direction as the group; and (iii) avoidance, the willingness to avoid collisions with others. Vicsek *et al.,*[2] later introduced a model based on statistical analysis, which averaged the velocities of individuals within a defined proximity of the individual and accounted for individual tendencies as stochastic noise. Couzin *et*

al.,[3] expanded upon these ideas by combining the rules defined by Reynolds with the area of interaction defined by Vicsek. Authors propose that each rule have an area of interaction: the closest avoidance, the following alignment and the biggest one attraction. These 3 rules effectively reproduce in simulation three collective behaviors observed in nature, swarming (no orientational order), schooling (highly polarized movement), and milling (circular, swirling movement). Models that follow these guides are now known as "3A models"[4, 5, 6].

Most of available approaches to 3A-models rely on artificial tools to describe or enforce the rules, such as inter-individual distance[7, 8, 9], global velocity[10, 11], or absolute communication among individuals[12, 13]. However, these methods do not accurately reflect the way animals interact with their environment or others. To address this gap, researchers have explored biologically plausible ("bio-plausible") ways of describe the environment[14, 15, 16, 17], with vision proving to be the most effective and widespread one[18, 19, 20, 21, 22]. Several vision-based models have been tested on robotic platforms[23, 24], but both of these either use vision merely as a selective mechanism[25, 26] or lack alignment with the fundamental principles of the 3A models[27].

Castro *et al.* (2024) [28] introduced a model that utilizes only visual information while maintaining the traditional 3A model structure. This model effectively replicates the phases of collective motion described by Couzin *et al.,* by employing a visual mechanism known as optic flow. Optic flow, which captures the apparent angular motion of the environment due to the relative movement between an observer and its surroundings, has been validated as a crucial navigational tool in nature. Optic flow magnitude is used by insects during most flight maneuvers [29, 30] as well as in fish and birds [31, 32]. Furthermore, several robotic platforms have already incorporated small sensors dedicated to measuring optic flow, highlighting its potential as a key mechanism for both biological understanding and innovation[33, 34, 35, 36, 37].

Building on these findings, this study proposes a "robot-in-the-loop" approach to collective motion that utilizes only visual information, implemented on a educational robotic platform. Unlike previous models, this approach does not rely on global information or complex communication networks. Instead, it introduces a visual anchor to account for environmental boundaries, thereby eliminating the need to model wall interactions explicitly. Experimental results and numerical simulations demonstrate that this method successfully reproduces several collective motion behaviors on a flock of 10 independently controlled robots.

The article is organized as follows: Section 2.1 provides an overview of the challenges of a robotic implementation of 3A-models, with the example of the visual-based collective motion model proposed by Castro *et al.* (2024) [28] (Fig.1). Section 2.2 describes the robotic platform and the model modifications for implementing on the 10 independently controlled robots including the introduction of the visual anchor concept. Section 2.4 detail the implementation of a feature matching optic flow and visual collective motion rules. Section 2.5 describes the supplementation on original model

and summarizes robotic setup, while Section 3 presents the results from simulations and robot-in-the-loop experiments. Finally, Section 4 concludes with a discussion of the results and directions for future research.

2. Methods

2.1. Robotic Implementation challenges

Flocking models that are based on individual rules of attraction, avoidance and alignment, or better known as 3A-models, present several challenges when being implemented on a robotic platform. The biggest two are related to how individuals are modeled. The assumption of most models is that each individual is a self propelled particle, traditionally moving on a plane. Yet most robots are particles or restrict their movement only to a plane. This brings forth the first challenge, which robot kinematics should be chosen? It should balance following the assumptions of the model, sensing capabilities, computational capacity, size, among others. By exiting the numerical environment, the experimental environment are not infinite anymore, so the collective behavior should be confined which is a second challenge. Without this considerations flocking on robots would be as direct as a raw implementation of a established model, such as Castro *et al.,* (2024) [28] on a the selected robotic platform (as seen on Fig. 1). The following sections will present our solution to both challenges.

2.2. Robotic setup

This section will section will present our robot setup to a robot-in-the-loop visual flocking model. We propose to use an underpowered, commercially available robotic platform that offloads the sensing capabilities to a base station but independently uses the information from its binary visual map to execute the model presented in the previous section.

This robot-in-the-loop solution utilizes an existing educational robotic platform, the Sphero Bolt®(Fig. 2a). These are small, spherical robots that can be controlled in real-time via Bluetooth®and an educational app, where users can set a heading value, desired velocity, LED states, and read the onboard sensors' values. Additionally, several open-source projects integrate these robots with different programming environments such as ROS, Python, or MATLAB®.

Each robot is equipped with several sensors, including an Inertial Measurement Unit (IMU) comprising a 3-axis accelerometer, gyroscope, and magnetometer, and a light sensor. It also features four infrared (IR) proximity sensors on its sides, two addressable RGB LEDs (one in the front and one in the back), and an 8x8 RGB LED matrix on top.

The Sphero's movement is achieved via a self-balancing two-wheel enclosed system, with two non-actuated wheels maintaining constant contact with the enclosing sphere.

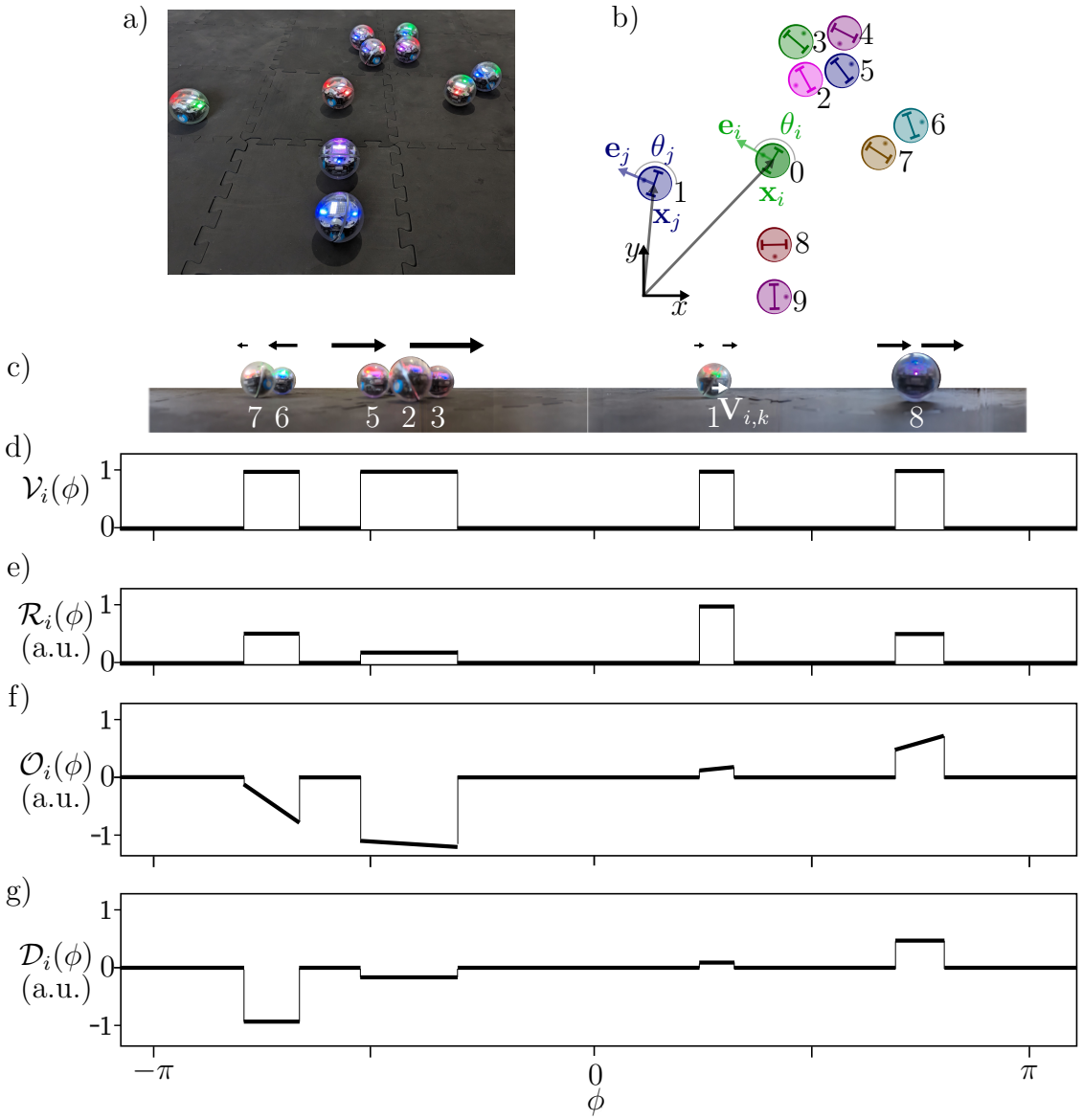

Figure 1. Model collective behaviors from optic flow and retinal cues [28]. a) View of the flock positions.b) Representation of the flock positions after recognition at time step t. Each i-particle is described by its XY-coordinates and unit heading vector $\mathbf{e}_i$, with the heading angle θ_i. c) Point-of-view of particle 0 showing the retinal relative positions of each particle. The arrows represent the perceived optic flow, and the white arrow indicates the apparent velocity ($\mathbf{V}_{i,k}$). Here, we only see the azimuthal component of the apparent velocity $\mathbf{V}_{i,k}$ by the particle i="0" of retinal object "k". d) $\mathcal{V}(\phi)$ binary panorama representation of the field of view of particle 0. e) Apparent distance $\mathcal{R}_i(\phi)$ of particle "0". f) Optic flow $\mathcal{O}_i(\phi)$ perceived by particle 1. g) Optic flow divergence $\mathcal{D}_i(\phi)$ perceived by particle 1.

This dual-motor design allows for stationary turning; however, while moving, the turning is achieved by differential motion (Fig. 2b-c). The locomotion generates a pitch displacement depending on the speed and the friction coefficient between the sphere's encapsulation and the surface (Fig. 2d). This pitch is partially managed by the robot's

dual-MCU configuration: one MCU handles the communication protocol and high-level instructions, while the other manages motor control to ensure the robot achieves the desired heading and speed, along with low-level sensor readings.

To address the lack of an onboard camera and the inability to embed one into the robots, we set up a bird's-eye camera above the experimental area. This is an industrial RGB global shutter camera, the *Daheng Imaging MER2-230-168U3C*, with a frame rate of 168 FPS, a resolution of 1920x1200, and support for interchangeable c-mount lenses (Table 1). The selected lens is a 4mm, 1/8", F2.0-12 non-distorting lens, which allows the camera to be mounted 2.3 m above the robots, covering a 2.7x2.7 m working area (approximately 38 times the length of the robots) (Fig. 3a).

Shutter time	25 ms
Pixel Bit Depth	10 bit
Digital gain	5 dB
Pixel Data Format	Bayer RG10
Lens f-stop	11
Interface	Python 3.10

Table 1. Camera Settings.

Each robot is assigned an individual color pattern that enables identification of the robot's front and back. To maintain light connectivity between the front and back, the back half of the LED matrix is set to 50% bright white light. This configuration provides 16 possible combinations for identifying each robot, with the potential for

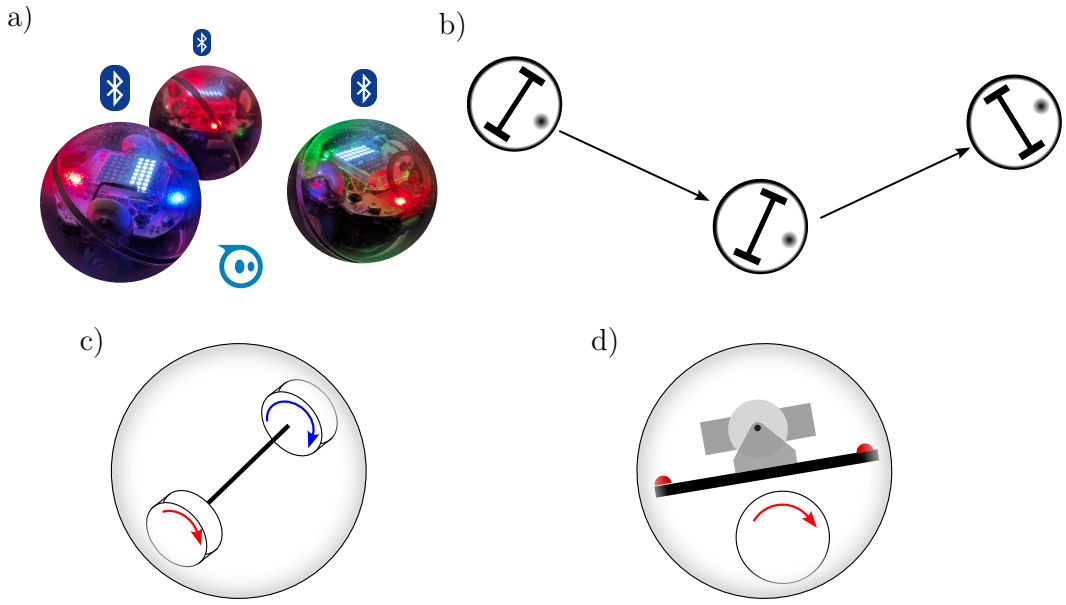

Figure 2. Robotic platform. a) Sphero Bolt®wireless educational robots. b) Example of the robots' movement, c) Heading control by differential rotation on the two opposite inner wheels, d) Induced nose-up pitch during forward displacement.

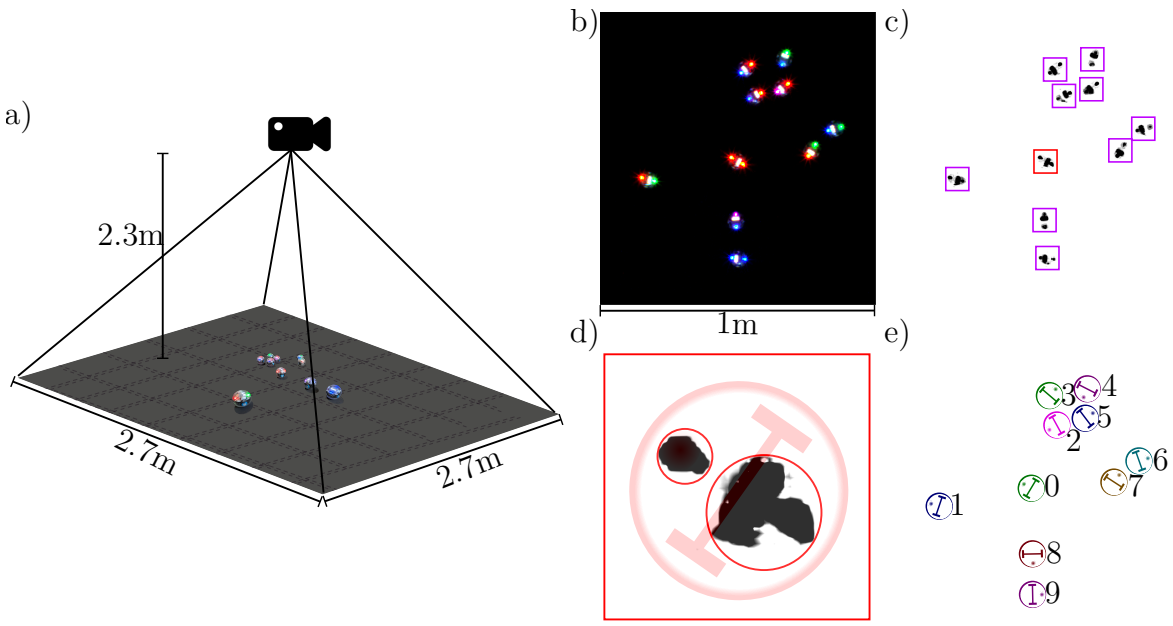

Figure 3. Experimental setup. a) Picture of the experimental setup, b) Example of the images taken by the camera (Daheng Imaging MER2-230-168U3C camera with 4mm c-mount lens), c) Whole robot body image segmentation, d) Individual light sources and heading recognition, e) Individual robot identification and virtualization.

expansion using additional colors. The bird's-eye camera allows the isolation of each light source on top of the robots, preventing light bleeding between robots when they are in close proximity (Fig. 3b). This is achieved by capturing a 1920x1200 RGB image and performing whole-robot segmentation. This segmentation treats the three light sources as a single entity, allowing calculation of the center's XY-coordinates, which approximate the robot's center. Knowing the radius of the robots in pixels allows for cropping of N images from these coordinates, each isolating a single robot. The cropped images are further segmented to identify the front and back LEDs, as well as the LED matrix. The larger area generated by the two light sources enables front and back recognition (Fig. 3d). Additionally, by correlating the color from this segmentation, each robot can be uniquely identified. The final step is to compute the robot's heading by tracing the vector between the center positions of the front and back LEDs. This process results in the identification, XY-coordinates, and heading of each robot (Fig. 3e).

The next step, after identifying each robot and determining their location and heading, is to reconstruct the binary panorama of each robot.

2.3. Anchoring the collective behavior

In the binary panorama reconstruction, we introduce a new component to the original model [28]: a visual anchor. The inclusion of an anchor bounds the flock to a place and prevent infinite expansion. The anchor is an augmented reality (AR) solution that enables each robot to see the desired center point (Fig. 4a). This anchor is omnisciently

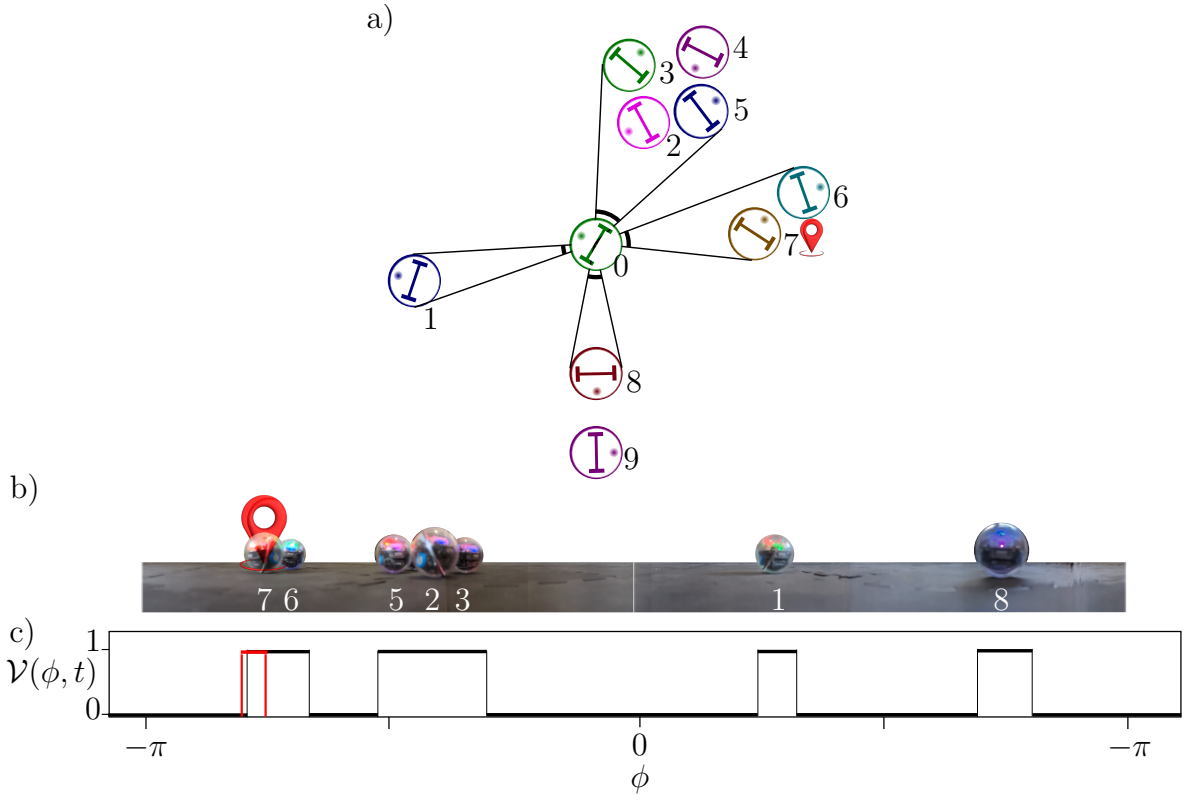

Figure 4. Binary panorama reconstruction. a) 2D virtual representation of the binary panorama, the red goal represents the desired location of the flock. b) Mixed reality point of view of the 0-particle with the omniscient red goal. c) Graph representation of all $\mathcal{V}(\phi, t)$. $\mathcal{V}(\phi, t)$ is a binary panorama representation of the field of view of particle i where the anchor is represented a different channel (red). This graph can be stored as a tuple list (rising and falling edges).

visible, regardless of any obstacles between the line of sight and the anchor location (Fig. 1b). It is a new visual channel that uses the same binary representation as the visual surroundings (Fig. 1c). The binary panorama is represented in Python as an ordered tuple list of every retinal object's rising and falling edge, with the last position always being the goal tuple. This process is performed independently for each particle without sharing information between particles.

2.4. Robotic Visual Flocking Model

Avoidance effect ($\mathcal{A}$) is here introduced in the attraction term ($\omega_{\odot}$). Our previous study [28] did not include any avoidance because most collective motion models are based on the assumption that all particles have some extra room on the z-axis, allowing them to avoid collisions with additional depth. This assumption is primarily based on how fish flocking data is gathered, where the fish can swim above or below each other in a few centimeters of water [38]. However, for any robotic implementation, obstacle or collision avoidance is of critical importance. Fortunately, avoidance and attraction have

similar characteristics, being proportional instantaneous responses to the optical size and location of a retinal object. The key difference is in the direction of the response: while attraction is positive, avoidance is negative. Therefore, avoidance is implemented by providing a new piecewise constant function ($\mathcal{A}$), where attraction is inverted in situations where a particle should avoid a retinal object. This new function depends on the width of the retinal object, specifically the difference between its rising and falling edges ($\Delta\phi$). This being a indirect measurement of the apparent distance ($\mathcal{R}(\phi, t)$). We choose $\mathcal{A}(\Delta\phi) = -5$ for $\Delta\phi < \pi/10$, which is equivalent to the retinal object being closer than three particle radii ($\mathcal{R} < 3a$). In all other cases, $\mathcal{A}(\Delta\phi) = 1$. The blind spot remains unchanged, giving more weight to the retinal object in front of the particle, regardless of this new function.

Alignment ($\omega_{\parallel}$) remains unchanged with respect to previous study [28], considering that the divergence/convergence of the optic flow already accounts for retinal objects approaching the focal object.

The optic flow for each retinal object can be decomposed in two components: the azimuthal in its center ($\overline{\mathcal{O}}$) and its radial divergence ($\mathcal{D}$). The former represents the apparent displacement of retinal features, while the latter represents changes in the size of those features. Divergence is a decrease in size, and convergence is an increase. With a 1D representation, these features are the rising and falling edges of the retinal object, akin to a feature-matching optic flow calculation.

Calculating optic flow as a feature-matching operation heavily relies on accurately identifying the same features across consecutive time steps. The model accomplishes this by finding the gradient of the Euclidean distance of the retinal object position and size between consecutive time steps. Regardless of flock size, there are three possible cases. If, in both time steps, there are the same number of retinal objects, the minimum of the gradient will be continuous and located on the main diagonal of a square matrix. When the number of retinal objects differs, it indicates either a separation or merge of retinal objects. In this case, there is no match, and the retinal objects are ignored in the calculation.

The apparent velocity vector $\mathbf{V}_{ik}$ in polar coordinates is $\mathbf{V}_{ik} = [-\mathcal{D}_i, \overline{\mathcal{O}}_i]\cdot\mathcal{R}_i/U$ with the heading vector $\mathbf{e}_i = [\cos\phi, -\sin\phi]$. The supplemented visual model equations for the collective behavior are presented in Eqs. 1*a*-*f*. Note that the noise k_η and the time delay τ are set to 0 for robot-in-the-loop experiments, as they are parameter modeling real-world noise and uncertainty.

$$\dot{\mathbf{x}}_i(t) = U\mathbf{e}_i(t), \tag{1a}$$

$$\dot{\theta}_i(t) = k_\odot\,\omega_\odot(t) + k_\parallel\,\omega_\parallel(t) + k_c\,\omega_c(t) + k_\eta\,\eta(t), \tag{1b}$$

$$\omega_\odot(t) = \left\langle \int_{-\pi}^{\pi} \mathcal{R}_i^2(\phi, t-\tau)\mathcal{A}(\mathcal{R}_i(t-\tau))b_\epsilon(\phi, t-\tau)\sin(\phi, t-\tau)\,d\phi \right\rangle, \tag{1c}$$

$$\omega_\parallel(t) = \left\langle \int_{-\pi}^{\pi} \frac{\mathbf{e}_i(t-\tau)\times\mathbf{V}_{ik}(t-\tau)}{U\mathcal{R}_i(\phi, t-\tau)} b_\epsilon(\phi, t-\tau)\,d\phi \right\rangle, \tag{1d}$$

$$\omega_c(t) = \int_{-\pi}^{\pi} \frac{\mathcal{R}_i^2(\phi, t-\tau)\sin(\phi, t-\tau)}{a} d\phi, \tag{1e}$$

$$\frac{\mathbf{e}_i(t) \times \mathbf{V}_{ik}(t)}{\mathcal{R}_i(t)} = -\mathcal{D}_i(\phi, t)\sin(\phi, t) + \overline{\mathcal{O}}_i(\phi, t)\cos(\phi, t). \tag{1f}$$

Lastly, the robotic platform introduces several critical processing factors – Bluetooth, image processing, internal Spheros control command, ... – that affect the overall dynamical response of the robots and its collective behavior as a whole (see Supplementary Information for more details). These extra factors results in a time delay τ with a median of 600ms. The overall functional diagram of the collective motion model is presented in Fig. 5. This model is executed independently and simultaneously for all robots; it does not involve communication between robots, sharing information, and only uses visual data. The same block could be utilized on a robotic platform using LiDAR, onboard cameras, touch sensors, ultrasound, or any other sensor that provides a binary visual panorama as output. All the information used by this model is self-referred and self-contained; there is no global reference frame, coordination, or knowledge of the flock or the environment.

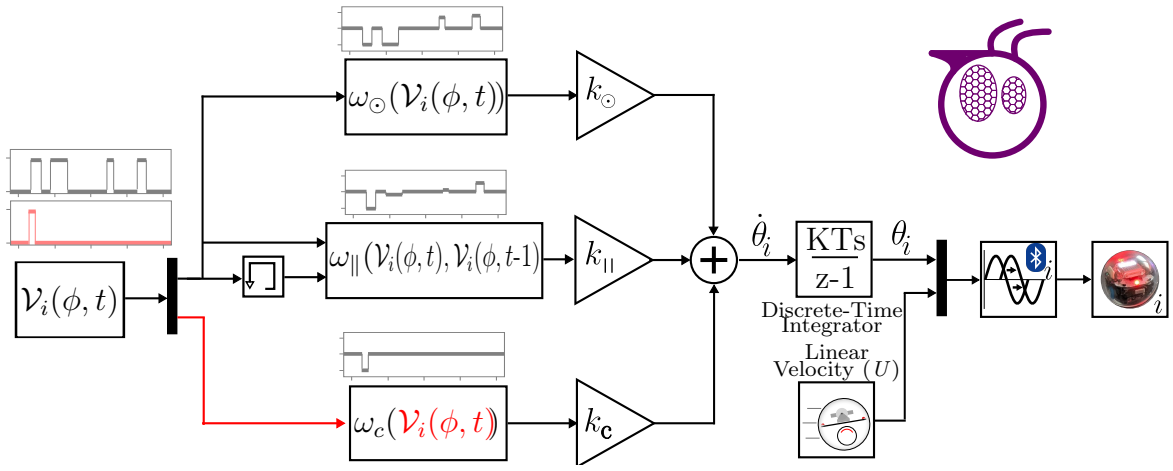

Figure 5. Functional diagram of the collective motion algorithm with robots-in-the-loop.

2.5. Robots-in-the-loop computer architecture

We propose a robots-in-the-loop flocking that consists of a set of N Sphero® Bolt robots, which are devoid of visual sensor and are controlled via Bluetooth®. Their positions and headings are calculated by processing the images from an industrial 1920x1200 RGB, 164 FPS overhead camera. From all their positions, a binary visual map is recreated. This visual map serves as the input for the collective motion model. This process is shown in Fig. 6.

There are two main Python programs to handle the implementation of these blocks. The first one (*administrator.py*) manages the camera communication, data

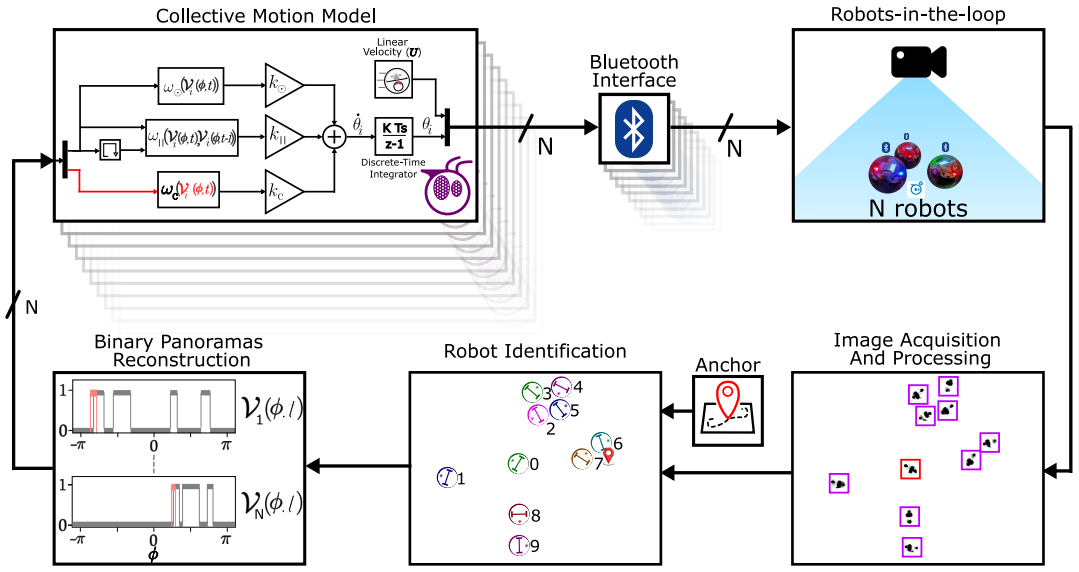

Figure 6. General Block diagram. The N flocking robots are seen by the 1920x1200px, 164fps camera. This image is acquired and processed to find the XY-coordinates, heading and colors for all N robots. With these coordinates each robot can be identified. The XY-coordinates of the Anchor can be set at any point of the image to be integrated to the identification of the robots. All these coordinates are processed to reconstruct the binary panorama for each of the spherical robots. Each binary Panorama is the input of independent Collective motion model that process the visual panoramic information. All these models are ran in parallel and their output in term of heading are sent to each robot via a Bluetooth interface.

extraction, image processing, object identification, and binary panorama reconstruction (Fig. 3). The output is transferred to N independent routines named *PyRobot.py* (the second program), which take a single binary panorama, perform the collective motion model, manage the Bluetooth connection, and send the desired heading and velocity (Fig. 5). Each robot receives the inputs (heading and velocity) and moves accordingly. Each *PyRobot.py* is a routine that can be executed either locally or remotely from the Administrator. This setup allows for scaling the flock size to the maximum number that can fit within the camera's recognition area. Note that this scaling is hardware-constrained by the Bluetooth interface capabilities of the computer. For *Windows*®machines, the theoretical maximum is seven devices per machine; in *Linux*, it is seven per interface. However, it is recommended to have a maximum of five on the same machine managing the camera and six per interface on independent machines.

2.6. Metrics

To quantitatively distinguish between different dynamical phases, we introduce three global order metrics: polarization (P), milling (M), and opacity (O) [39, 40]. These metrics are defined as follows:

$$P(t) = \left\| \overline{\mathbf{e}_i(t)} \right\|, \tag{2}$$

$$M(t) = \left\| \overline{\mathbf{y}_i(t) \times \mathbf{e}_i(t)} \right\|, \tag{3}$$

$$O(t) = \frac{1}{2\pi} \overline{\int_{-\pi}^{\pi} \mathcal{V}_i(\phi, t)\, d\phi}, \tag{4}$$

where the overbar denotes an average over all individuals, and the unit vector $\mathbf{y}_i(t) = \frac{\mathbf{x}_i(t) - \overline{\mathbf{x}_i}(t)}{\|\mathbf{x}_i(t) - \overline{\mathbf{x}_i}(t)\|}$ points towards particle i from the center of mass. All three metrics range within the interval $[0, 1]$. The polarization $P(t)$ measures alignment: $P(t) = 0$ corresponds to particles pointing in all directions, while $P(t) = 1$ represents a perfectly aligned school. The milling $M(t)$ quantifies the normalized angular momentum: $M(t) = 0$ corresponds to a straight-line formation, and $M(t) = 1$ represents perfect milling. The opacity $O(t)$ measures the "occupancy" of the visual fields: $O(t) = 0$ indicates no object in the visual field, and $O(t) = 1$ signifies that the entire visual field is obscured.

3. Results

3.1. Simulation of supplemented visual model

To make the problem dimensionless, we chose $a = 1$ and $U = 1$. We also kept the blind angle parameter and noise strength with values that lead to the most phase reproduction: $\epsilon = 1$, $k_\eta = 0.01$. With this approach, three dimensionless parameters remain: the strengths of anchor attraction and alignment, k_c, $k_\odot$, and $k_\parallel$. The anchor is a parameter aiming for a contained robotic experimentation, set practically to the minimum value that ensures the robots do not reach the arena's border when no other law is in effect, $k_c = 0.005$.

The visual original model has been supplemented with three major changes impacting the phase distribution and the metric on the parameter space $(k_\odot, k_\parallel)$:

- the inclusion of the avoidance rule ($\mathcal{A}(\phi)$) (Fig. S7),
- the addition of an anchor law (ω_c) (Fig. S8),
- the introduction of a time delay (τ) (only in simulation) to reflect the time delay between the desired heading and a perceived change in heading while operating the Spheros robot, (Fig. 7).

We now compare the original model with each consecutive addition by exploring the parameter space $(k_\odot, k_\parallel) \in [0, 0.3] \times [0, 0.3]$ on a flock of 10 individuals. For each parameter set, we ran 10 simulations over a long duration ($\Delta t = 5000$). The mean values of P, M, and O were determined by averaging over the last 1000 time units to ensure that transients do not influence the results. The outcomes of these simulations are synthesized in phase diagrams (Fig. 8).

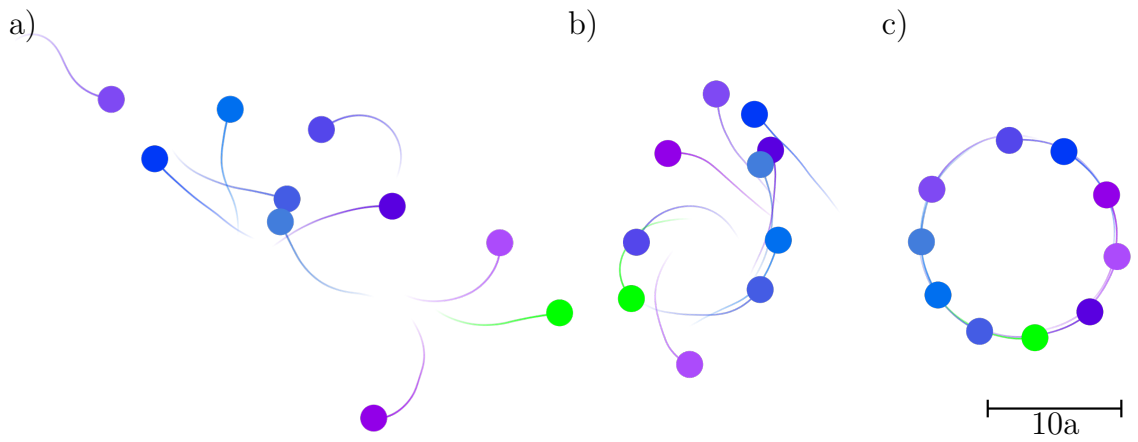

Figure 7. Illustration of the phases observed in simulation for the supplemented visual model with the addition of avoidance, anchor, and time delay as defined in Eqs. 1*a*, with $\tau = 2.4[a/U]$. Three phases are observed when varying the other two parameters: a) Swarming ($k_\odot = 0.25$, $k_\parallel = 0.04$) (see SVideo 1); b) Bistable ($k_\odot = 0.2$, $k_\parallel = 0.25$) (see SVideo 2); c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.16$) (see SVideo 3).

We maintain the identification of the collective phases from the values of the polarization P and milling M parameters: schooling when $P > 0.5$ and $M < 0.5$, milling when $P < 0.5$ and $M > 0.5$, and swarming when $P < 0.5$ and $M < 0.5$.

3.2. Phase diagrams of the supplemented model in simulation

We now compare the model with each consecutive addition and the robotic implementation, exploring the parameter space $(k_\odot, k_\parallel) \in [0, 0.3] \times [0, 0.3]$ for a flock of 10 independently controlled robots. For the robotic implementation, the dimensionless parameters are set to match the radius in pixels and velocity in radius per second, resulting in $a = 1$, with the other parameters remaining the same ($\epsilon = 1$, $k_\eta = 0$, and $k_c = 0.005$). For each parameter set, we ran 5 robotic experiments over a long duration (240sec or $\Delta t \approx 8500$). The mean values of P, M, and O were determined by averaging over the last 1000 time units to ensure that transients do not influence the results. The outcomes of these experiments are synthesized in phase diagrams (Fig. 9a). An example of swarming is shown in Fig. 9b, of partial schooling for a bistable phase in Fig. 9c, and of milling in Fig. 9d.

The phase diagram of the original model [28] showcases the reproduction of the main three phases (swarming, schooling, and milling), with the area above $k_\odot > 0.15$ and $k_\parallel > 0.15$ being bistable (Fig. 8a). A bistable phase oscillates between milling and schooling (Fig. 8b). By adding repulsion to the model, this bistable phase is removed, but the area where milling appears shrinks to roughly half of the original while the final opacity of the flocks is reduced. Opacity is an inverse metric of flock density, and this reduction aligns with the expected effect of adding avoidance.

Fig. 8c shows the introduction of the anchor, which eliminates the schooling phase while the swarming phase area remains unchanged. This change is attributed to the

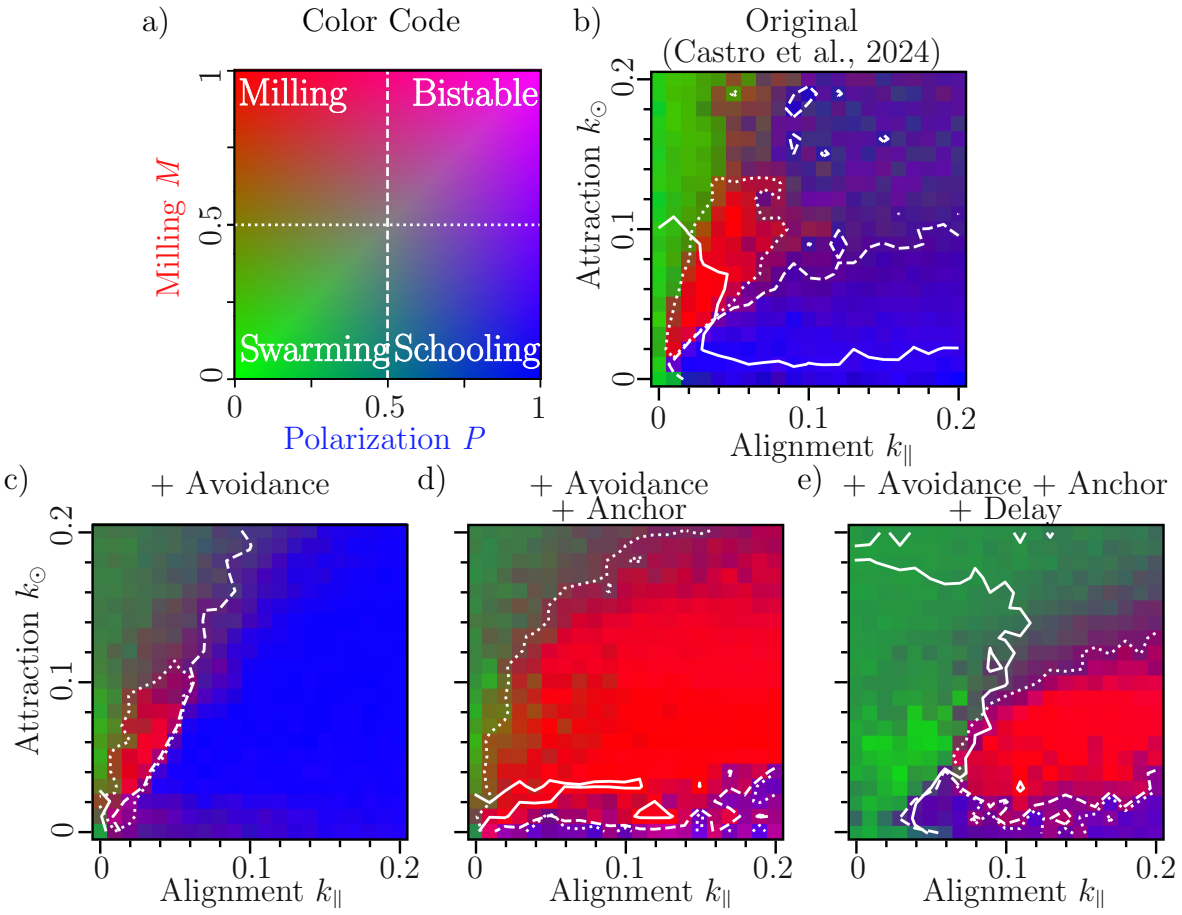

Figure 8. Simulation Phase Diagrams for 10 individuals. The dotted line is $P = 0.5$, the dashed line is $M = 0.5$, and the solid line is $O = 0.3$. A) Color code, b) Original visual model [28], c) Supplemented visual model with avoidance term, $\mathcal{A}(\mathcal{R}(\phi, t))$, d) Supplemented model with avoidance and anchor term, $\omega_c(t)$, e) Supplemented visual model with avoidance, anchoring and perceived time delay (τ=2.4[a/U]).

turning effect generated by the anchor. When all particles are in a line, the front particles turn faster than those at the back, resulting in a turn that eventually stabilizes into a circle and a very stable milling pattern.

Adding a time delay ($\tau = 2.4[a/U]$) in the supplemented visual model (Fig. 8d) causes the swarming phase to expand to values above $k_\odot = 0.17$ or when $k_\parallel < 0.08$. For low attraction ($k_\odot < 0.03$), there is bistability, and the milling phase is reduced to an area for alignment above $k_\parallel = 0.1$ and medium attraction ($0.06 < k_\odot < 0.17$).

3.3. Comparison with the phase diagram of the Spheros' robots-in-the-loop

For the robotic implementation (see Fig. 9), we found that the milling phase remains fairly stable, appearing for high alignment ($k_\parallel > 0.15$) across most attraction values ($k_\odot < 0.16$). The swarming phase is constant for low alignment ($k_\parallel < 0.06$) or high attraction ($k_\odot > 0.17$).

The avoidance law, subject to the same time delay, results in some collisions that

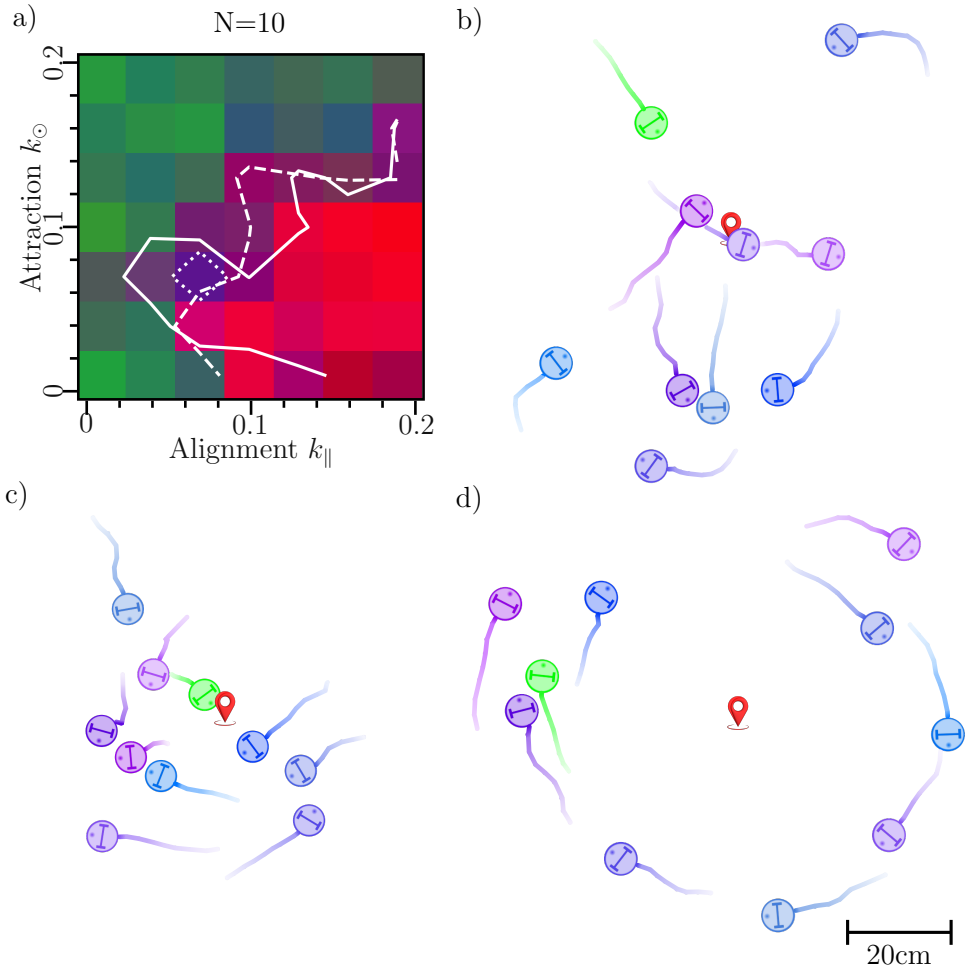

Figure 9. Experimental results for a flock of 10 independently controlled robots (with $U = 4a/sec = 0.14[m/sec]$ (the radius of Spheros robot is $a = 3.5cm$). a) Phase diagram, the dotted line is $P = 0.5$, the dashed line is $M = 0.5$, and the solid line is $O = 0.7$. b) Swarming ($k_\odot = 0.25$, $k_\parallel = 0.04$) (see raw footage SVideo 4, robotic schematic SVideo 5 or a 3rd person view SVideo 6); c) Bistable ($k_\odot = 0.2$, $k_\parallel = 0.25$) (see raw footage SVideo 7, robotic schematic SVideo 8 or a 3rd person view SVideo 9); d) Milling ($k_\odot = 0.07$, $k_\parallel = 0.16$) (see raw footage SVideo 10, robotic schematic SVideo 11 or a 3rd person view SVideo 12).

resolve over time, leading to higher opacity values. Similarly, for $k_\odot < 0.01$, avoidance is too small, and the flock groups around the anchor without moving. This behavior is somewhat mitigated when there is enough alignment ($k_\parallel > 0.15$), but the flock remains more compact than in the simulations.

4. Discussion

The collective motion model with the robotic platform reproduces several collective patterns such as swarming, milling, and a bistable phase. For the first time to the best of our knowledge, a visual flocking on spherical robots shows a phase diagram that

matches almost perfectly that of the simulation with the same visual model (Fig. 8e *vs* Fig. 9a). The mean squared error (MSE) between the phase diagram images of robots-in-the-loop and supplemented visual model results is 2.48% (for more details see Fig. S2). Such implementation therefore bridges the gap between collective model simulation and real-world experiments, allowing for expansion to N individuals with low complexity. A model based on visual information enables a mixed-reality approach where there can be interaction between the flock and the environment without adding complexity to the model or sensors available to each robot.

Future works with these spherical robots could include an explicit channel for avoidance also using optic flow, a moving anchor, a replacement of the anchor by another particle without adding a new channel, and an expansion of the flock size to 20 or 40 individuals for a closer-to-nature implementation of these visual collective models.

References

[1] Reynolds C W 1987 Flocks, herds and schools: A distributed behavioral model *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* pp 25–34

[2] Vicsek T, Czirók A, Ben-Jacob E, Cohen I and Shochet O 1995 *Physical review letters* **75** 1226

[3] Couzin I D, Krause J, James R, Ruxton G D and Franks N R 2002 *Journal of theoretical biology* **218** 1–11

[4] Mishra S, Baskaran A and Marchetti M C 2010 *Physical Review E* **81** 061916

[5] Solon A P, Caussin J B, Bartolo D, Chaté H and Tailleur J 2015 *Physical Review E* **92** 062111

[6] Vicsek T and Zafeiris A 2012 *Physics reports* **517** 71–140

[7] Chaté H, Ginelli F, Grégoire G and Raynaud F 2008 *Phys. Rev. E* **77**(4) 046113 URL `https://link.aps.org/doi/10.1103/PhysRevE.77.046113`

[8] Bialek W, Cavagna A, Giardina I, Mora T, Silvestri E, Viale M and Walczak A M 2012 *Proceedings of the National Academy of Sciences* **109** 4786–4791

[9] Wen G, Duan Z, Li Z and Chen G 2012 *International Journal of Robust and Nonlinear Control* **22** 1790–1805

[10] Cavagna A, Del Castello L, Giardina I, Grigera T, Jelic A, Melillo S, Mora T, Parisi L, Silvestri E, Viale M *et al.* 2015 *Journal of Statistical Physics* **158** 601–627

[11] Zhang H T, Zhai C and Chen Z 2010 *IEEE Transactions on Automatic Control* **56** 430–435

[12] Shang Y and Bouffanais R 2014 *Scientific reports* **4** 1–7

[13] Virágh C, Vásárhelyi G, Tarcai N, Szörényi T, Somorjai G, Nepusz T and Vicsek T 2014 *Bioinspiration & biomimetics* **9** 025012

[14] Ginelli F and Chaté H 2010 *Physical review letters* **105** 168103

[15] Nagy M, Ákos Z, Biro D and Vicsek T 2010 *Nature* **464** 890–893

[16] Lazarus J 1979 *Animal Behaviour* **27** 855–865

[17] Bode N W, Franks D W and Wood A J 2011 *Journal of The Royal Society Interface* **8** 301–304

[18] Lafoux B, Moscatelli J, Godoy-Diana R and Thiria B 2023 *Communications Biology* **6** 585

[19] Krongauz D L, Ayali A and Kaminka G A 2024 *PLOS Computational Biology* **20** e1011796

[20] Strandburg-Peshkin A, Twomey C R, Bode N W, Kao A B, Katz Y, Ioannou C C, Rosenthal S B, Torney C J, Wu H S, Levin S A *et al.* 2013 *Current Biology* **23** R709–R711

[21] Davidson J D, Sosna M M, Twomey C R, Sridhar V H, Leblanc S P and Couzin I D 2021 *Journal of the Royal Society Interface* **18** 20210142

[22] Soria E, Schiano F and Floreano D 2021 *Nature Machine Intelligence* **3** 545–554

[23] Schilling F, Soria E and Floreano D 2022 *IEEE Access* **10** 28133–28146

[24] Mezey D, Bastien R, Zheng Y, McKee N, Stoll D, Hamann H and Romanczuk P 2024 *arXiv preprint arXiv:2406.17106*
[25] Soria E, Schiano F and Floreano D 2019 The influence of limited visual sensing on the reynolds flocking algorithm *2019 Third IEEE International Conference on Robotic Computing (IRC)* (IEEE) pp 138–145
[26] Pearce D J G, Miller A M, Rowlands G and Turner M S 2014 *Proceedings of the National Academy of Sciences* **111** 10422–10426 URL `https://www.pnas.org/doi/abs/10.1073/pnas.1402202111`
[27] Bastien R and Romanczuk P 2020 *Science advances* **6** eaay0792
[28] Castro D, Ruffier F and Eloy C 2024 *Phys. Rev. Res.* **6**(2) 023016 URL `https://link.aps.org/doi/10.1103/PhysRevResearch.6.023016`
[29] Franceschini N, Ruffier F and Serres J 2007 *Current Biology* **17** 329–335
[30] Portelli G, Ruffier F and Franceschini N 2010 *Journal of Comparative Physiology A* **196** 307–313
[31] Serres J R, Evans T J, Åkesson S, Duriez O, Shamoun-Baranes J, Ruffier F and Hedenström A 2019 *Journal of the Royal Society Interface* **16** 20190486
[32] Bonnen K 2023 *Current Biology* **33** R30–R32
[33] Ruffier F and Franceschini N 2005 *Robotics and Autonomous Systems* **50** 177–194
[34] Expert F and Ruffier F 2015 *Bioinspiration & biomimetics* **10** 026003
[35] Floreano D, Pericet-Camara R, Viollet S, Ruffier F, Brückner A, Leitel R, Buss W, Menouni M, Expert F, Juston R *et al.* 2013 *Proceedings of the National Academy of Sciences* **110** 9267–9272
[36] Bergantin L, Harbaoui N, Raharijaona T and Ruffier F 2021 *Journal of the Royal Society Interface* **18** 20210567
[37] De Croon G C, Dupeyroux J J, De Wagter C, Chatterjee A, Olejnik D A and Ruffier F 2022 *Nature* **610** 485–490
[38] Rosenthal S B, Twomey C R, Hartnett A T, Wu H S and Couzin I D 2015 *Proceedings of the National Academy of Sciences* **112** 4690–4695
[39] Calovi D S, Lopez U, Ngo S, Sire C, Chaté H and Theraulaz G 2014 *New journal of Physics* **16** 015026
[40] Filella A, Nadal F, Sire C, Kanso E and Eloy C 2018 *Physical review letters* **120** 198101
[41] Castro D, Eloy C and Ruffier F 2024 Complete Data Deposit for: Visual collective behaviors on spherical robots URL `https://doi.org/10.57745/12N99K`
[42] Castro D, Eloy C and Ruffier F 2024 Complete data, media and code deposit of: Visual collective behaviors on spherical robots URL `osf.io/ugwr9`

Acknowledgements

The participation of D.C. in this research was made possible by joint PhD grant from Aix-Marseille University.
D.C. and F.R. were also supported by Aix Marseille University and the CNRS (Life Science, Information Science, and Engineering and Science & technology Institutes).
The facilities for the experimental tests has been mainly provided by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015).

Data Availability

Data acquired during the experiments and simulations are available online [41, 42]

Video Accessibility

Links to supplemental videos are provided in the supplementary information and can also be found at: youtube or [42].

Author contributions statement

D.C. performed the simulations, the robotic experiments, and data consolidation. C.E. & F.R. outlined the robot in the loop experiment and supervised its implementation. All authors: Proofread the manuscript.

Supplementary information of Visual collective behaviors on spherical robots

Diego Castro[1,2],Christophe Eloy[2] and Franck Ruffier[1,*]

[1]Aix Marseille Université, CNRS, ISM, Marseille, 13288, France
[2]Aix Marseille Université, CNRS, Centrale Med, IRPHE, Marseille, 13013, France
[*]Author to whom any correspondence should be addressed.

E-mail: franck.ruffier@cnrs.fr

Submitted to: *Bioinspir. Biomim.*
26 September 2024

S1 Spheros' dynamical response and time delay τ estimation

The robotic setup is composed of several steps, from setting a heading and velocity, to recognizing a XY-coordinate and heading from the image. Each step involves their own input treatment, response time and processing. As such, we performed a step response analysis for both heading and velocity independently. This analysis allows to see if there are what are significant changes from the intimidate response of the simulation.

Fig. S1 a, shows the average step response for a change in the heading of the robot. The change ranged from 1 to 180 degrees. Fig. S1 b, shows the average step response for a change in the velocity of the robot. The change ranged from 1 to 25 a/sec. Fig. S1 c, shows the average step response for a change in the yaw $\dot{\theta}$ of the robot. Fig. S1 d-e, shows this speeds ramps in the measured variables. All steps responses were performed 5 times with 10 different robots. The Y-axis is a percentage of the desired heading. Finally, Fig. S1 f, shows the measurement sensitivity to changes in heading and Fig. S1 g, shows the error in following a desired yaw $\dot{\theta}$. These results show that the system has a median time delay τ of approximately 600ms in responding to any input signal change. The image processing takes from 10-50ms, the image acquisition take from 25-50ms. Leaving at best 400ms delay to the communication between the control and the robot, the inner processing and the actuation of the input. Therefore, the visual input for the collective motion model must reflect this transport delay.

S2 Direct comparison between Supplemented visual model and robot-in-the-loop Experiments

The difference between Phase Diagrams constructed from simulated and robots-in-the-loop experiments for 10 individuals is very marginal Fig. S2. The mean squared error

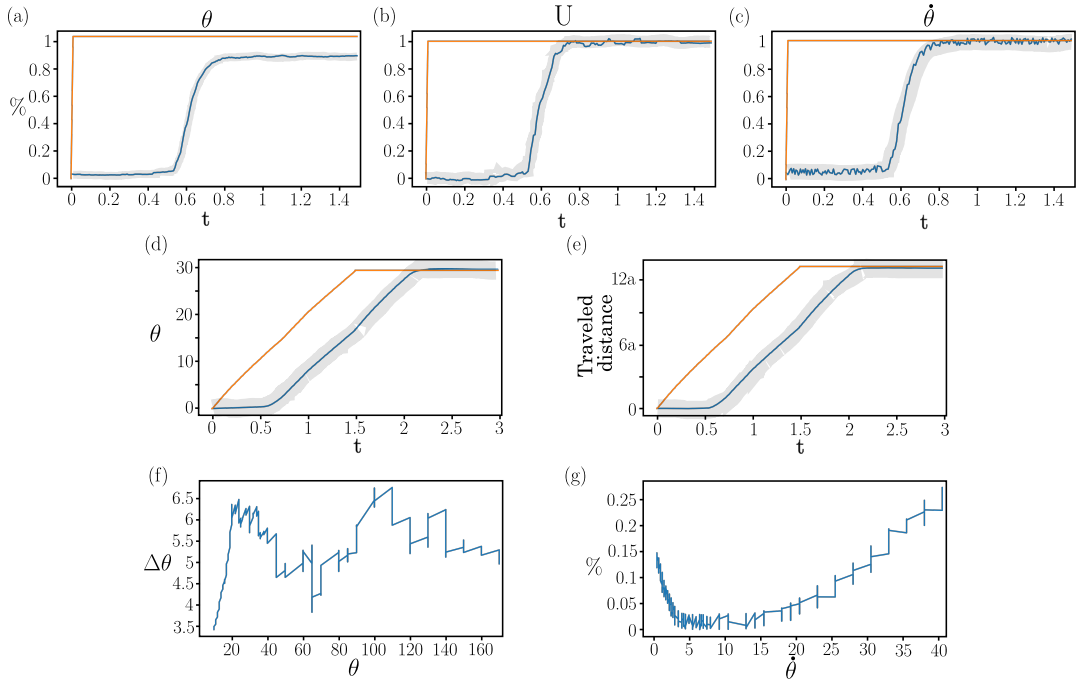

Fig. S1. Spheros' dynamical response (The grey shade is the average of normalized standard deviation) a) Average normalized heading θ step response of the 520 experiments. b) Average normalized velocity U step response [a/sec] of the 220 experiments. c) Average normalized yaw $\dot{\theta}$ step response [degrees] of the 220 experiments. d) Average heading θ ramp response [degrees/sec] of the 220 experiments. e) Average velocity step response as seen in traveled distance (with $a = 3.5cm$) of the 220 experiments. f) Heading θ measurement average sensitivity in degrees of the 520 experiments. g) Yaw $\dot{\theta}$ measurement average sensitivity in percentage of the 520 experiments.

– MSE – between the phase diagram images of robots-in-the-loop and supplemented visual model is 2.48%.

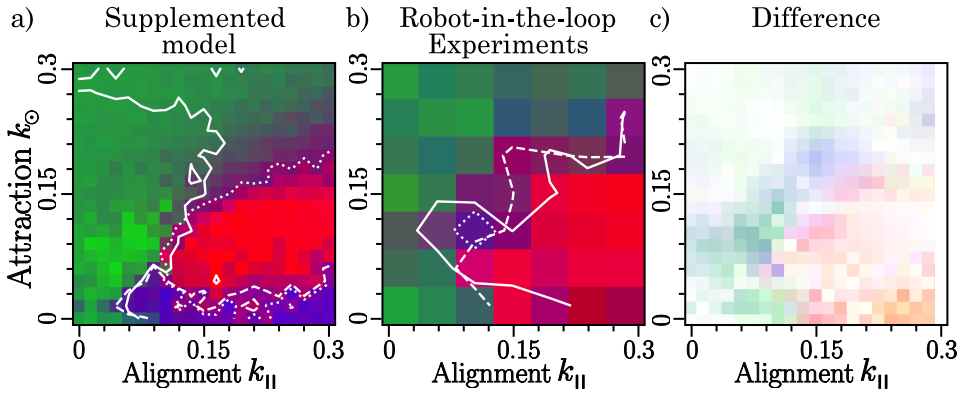

Fig. S2. Difference between the Phase Diagrams. a) Supplemented visual model (10 individuals), b) Phase Diagram of Spheros' robots-in-the-loop (10 robots), c) Raw difference image difference between both phase diagram. The mean squared error – MSE – between the phase diagram images of robots-in-the-loop and supplemented visual model is 2.48%.

S3 Image Processing Technical information

The camera details are presented on Table S1 The detailed dual image segmentation

Resolution	1920x1200
Frame rate	168fps
Pixel Size	5.86uM
Sensor Type	Sony IMX174
Optical Size	1/1.2”
Shutter Type	Global Shutter
Shutter time	52us 1s
ADC Bit Depth	10bit
Pixel Bit Depth	8bit, 10bit
Digital gain	0dB 24dB
Pixel Data Format	Bayer RG8 / Bayer RG10
Lens mount	type-C
Power Consumption	¡2.7W@5V

Table S1. Camera: *Daheng Imaging MER2-230-168U3C* specifications.

example is presented on Fig. S3.

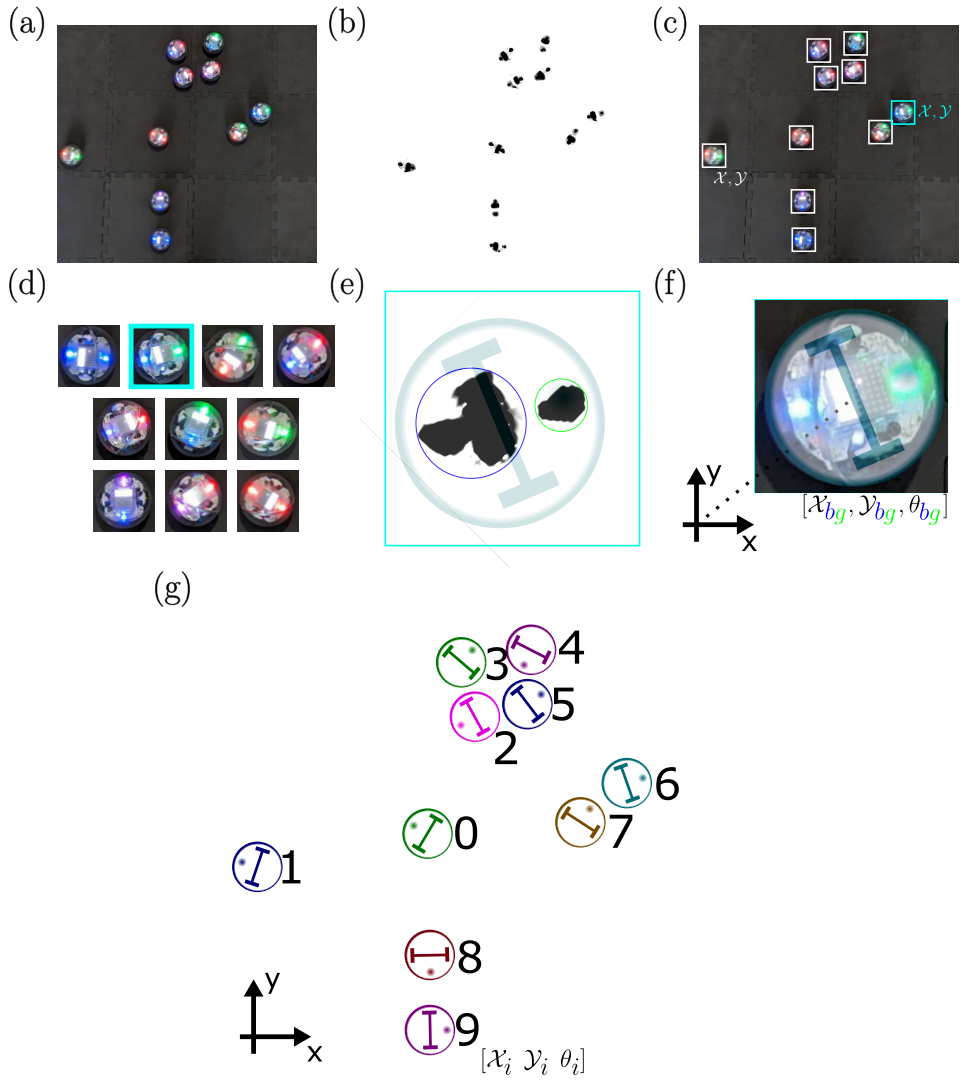

Fig. S3. Illustration of the phases observed for the original model. We observe three phases when varying the other two parameters: (a) Swarming ($k_\odot = 0.15$, $k_\parallel = 0$); (b) Schooling ($k_\odot = 0.07$, $k_\parallel = 0.3$); (c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.05$).

Feature Matching Optic Flow

Fig. S4 showcases examples of what real world each of the components of the optic flow would look like. It also correlated each with a binary panorama equivalency. Finally it shows a mixed case that would be a normal optic flow generation for both image and 1D cases.

For the feature matching optic flow the gradient generated when the amount of retinal object differ between consecutive time steps is shown on Fig. S5.

S4 Digital implementation of the supplemented visual model

The model presented on Eqs. 1, are the integral form of the model that use of the visual panorama. $\mathcal{V}_i(\phi, t-\tau)$ is a piece-wise constant function that can be expressed as a set unit boxcar functions (one for each retinal object).

Each retinal object can be described as a rising $\varphi_{\Uparrow k}$ and its corresponding falling edge $\varphi_{\Downarrow k}$. Furthermore, from these edges, we can compute the shade's mid-point $\varphi_k = (\varphi_{\Uparrow k} + \varphi_{\Downarrow k})/2$, and the retinal's object half-width $\Delta\varphi_k = (\varphi_{\Downarrow k} - \varphi_{\Uparrow k})/2$. The apparent distance, $\mathcal{R}_i(\phi, t-\tau)$ is defined as $\mathcal{V}_i(\phi, t-\tau)$ weighted by $\frac{a}{\sin\Delta\varphi_k}$.

This means that $\mathcal{R}_i(\phi, t-\tau)$ for a binary $\mathcal{V}_i(\phi, t-\tau)$, is a function independent of ϕ and is only be dependent on the k-retinal's object half-width ($\mathcal{R}_k(\Delta\varphi_k)$). Similarly, the avoidance function can be expressed as a function of $\Delta\varphi_k$, and is independent of ϕ. We add the subscript k, which identifies a single retinal object on the binary panorama of the individual i. Note that a retinal object k is different from an individual j, as in a retinal object there can be occlusion or aggregation of any j-particles on the i-th binary panorama. Then, we can rewrite Eq. 1c as:

$$\omega_\odot(t) = \left\langle \sum_k \mathcal{R}_i^2(\Delta\varphi_k, t-\tau)\mathcal{A}(\Delta\varphi_k, t-\tau) \int_{\varphi_{\Uparrow k}}^{\varphi_{\Downarrow k}} b_\epsilon(\phi, t-\tau)\sin(\phi, t-\tau)\, d\phi. \right\rangle \tag{S1}$$

where $b_\epsilon(\phi, t-\tau) = 1 + \epsilon\cos(\phi, t-\tau)$. And developing the integral we get

$$\omega_\odot(t) = \langle \sum_k 2\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)\mathcal{A}(\Delta\varphi_k, t-\tau)\sin(\Delta\varphi_k, t-\tau)\sin(\varphi_k, t-\tau) \\ (1 + \epsilon\cos(\varphi_k, t-\tau)\cos(\Delta\varphi_k, t-\tau))\rangle \tag{S2}$$

And here $\mathcal{R}_i(\Delta\varphi_k, t-\tau) = \frac{a}{\sin(\Delta\varphi_k, t-\tau)}$, Resulting in:

$$\omega_\odot(t) = \left\langle \sum_k \frac{2a^2\left(1 + \epsilon\cos(\varphi_k, t-\tau)\cos(\Delta\varphi_k, t-\tau)\right)}{\sin(\Delta\varphi_k, t-\tau)} \sin(\varphi_k, t-\tau) \right\rangle \tag{S3}$$

By substituting the $\langle\cdot\rangle$ normalization the implemented equation is:

$$\omega_\odot(t) = \frac{\sum_k \frac{2a^2(1+\epsilon\cos(\varphi_k, t-\tau)\cos(\Delta\varphi_k, t-\tau))}{\sin(\Delta\varphi_k, t-\tau)} \sin(\varphi_k, t-\tau)}{\sum_k \frac{2a^2(1+\epsilon\cos(\varphi_k, t-\tau)\cos(\Delta\varphi_k, t-\tau))}{\sin(\Delta\varphi_k, t-\tau)}} \tag{S4}$$

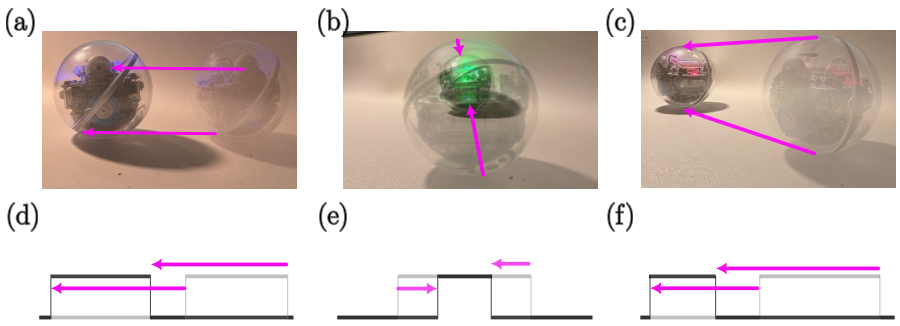

Fig. S4. Optic flow. a) Purely translational, b) Purely Convergent, c) Mixed Case, d) 1-D feature matching purely translational, d) 1-D feature matching purely convergent, d) 1-D feature matching mixed case.

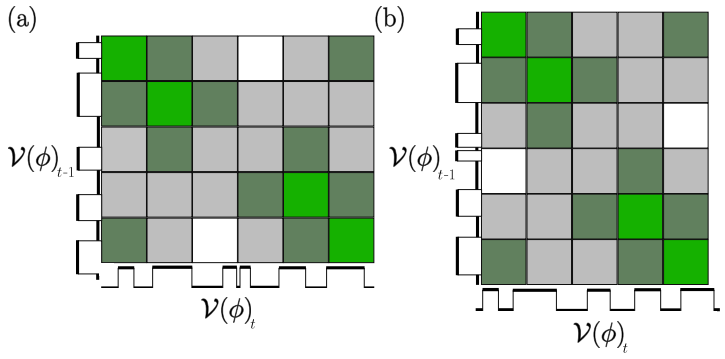

Fig. S5. Retinal object matching between consecutive time steps. a) Separation of retinal object, b) Merging of retinal objects.

The next step is to do the same for the alignment component. First, we replace Eq. 1f in Eq. 1d resulting in:

$$\omega_{\parallel}(t) = \int_{-\pi}^{\pi} U^{-1}(\overline{\mathcal{O}}_i(\phi, t-\tau)\cos(\phi, t-\tau) - \mathcal{D}_i(\phi, t-\tau)\sin(\phi, t-\tau))b_\epsilon(\phi, t-\tau)d\phi \quad \text{(S5)}$$

Note, that the visual alignment can also be written in polar coordinates (ϕ, r), with an azimuthal and radial component as follows:

$$\omega_{\parallel}(t) = \langle \omega_{\parallel,\phi}(t) + \omega_{\parallel,r}(t) \rangle \quad \text{(S6)}$$

$\mathcal{D}_i(\phi, t-\tau)$ and $\overline{\mathcal{O}}_i(\phi, t-\tau)$, similarly to $\mathcal{R}_i(\phi, t-\tau)$, are piece-wise constant function which are independent of ϕ but dependent on $\Delta\varphi_k$.

Taking Eq. S5 and Eq. S6:

$$\omega_{\parallel,\phi}(t) = \sum_k \mathcal{O}_i(\Delta\varphi_k, \phi, t-\tau)\left(\int_{\varphi_{\Uparrow k}}^{\varphi_{\Downarrow k}} \cos(\phi, t-\tau)b_\epsilon(\phi, t-\tau)d\phi\right) \quad \text{(S7)}$$

$$\omega_{\parallel,r}(t) = \sum_k -\mathcal{D}_i(\Delta\varphi_k, \phi, t-\tau)\left(\int_{\varphi_{\Uparrow k}}^{\varphi_{\Downarrow k}} \sin(\phi, t-\tau)b_\epsilon(\phi, t-\tau)d\phi\right) \quad \text{(S8)}$$

Eq. S8 is the exact form as Eq. S1 with the weight of the integral being $\mathcal{D}_i(\Delta\varphi_k, t-\tau)$ instead of $\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)\mathcal{A}(\Delta\varphi_k, t-\tau)$ resulting in:

$$\omega_{\parallel,r}(t) = \sum_k \frac{-2\mathcal{D}_i(\Delta\varphi_k, t-\tau)}{\mathcal{R}_i(\Delta\varphi_k, t-\tau)}\sin(\varphi_k, t-\tau)\left(1 + \epsilon\cos(\varphi_k, t-\tau)\cos(\Delta\varphi_k, t-\tau)\right) \quad \text{(S9)}$$

As for Eq. S7 the developing the integral and assuming $\Delta\phi \approx \sin\Delta\phi$ we have:

$$\omega_{\parallel,\phi}(t) = \sum_k \frac{\mathcal{O}_i(\Delta\varphi_k, t-\tau)}{\mathcal{R}_i(\Delta\varphi_k, t-\tau)}(2\cos(\varphi_k, t-\tau)(1 + \epsilon\cos(\Delta\varphi_k, t-\tau)\cos(\varphi_k, t-\tau)) + \epsilon(1 - \cos(\Delta\varphi_k, t-\tau))) \quad \text{(S10)}$$

Assumming that $\cos\Delta\varphi_k \approx 1$ and using Eq. 1f, then Eqs. S9-S10 become:

$$\omega_{\parallel}(t) \approx \left\langle \sum_k 2\frac{\mathbf{e}_i(t-\tau) \times \mathbf{V}_{ik}(t-\tau)}{U\mathcal{R}_i^2(\Delta\varphi_k, t-\tau)} b_\epsilon(\varphi_k, t-\tau) \right\rangle \quad \text{(S11)}$$

Susbtituting the normalization and the blindspot, Eq. S11 becomes:

$$\omega_{\parallel}(t) \approx \frac{\sum_k 2\frac{\mathbf{e}_i(t-\tau)\times\mathbf{V}_{ik}(t-\tau)}{U\mathcal{R}_i^2(\Delta\varphi_k,t-\tau)}(1+\epsilon\cos(\varphi_k,t-\tau))}{\sum_k 2\frac{\|\mathbf{V}_{ik}(t-\tau)\|}{U\mathcal{R}_i^2(\Delta\varphi_k,t-\tau)}(1+\epsilon\cos(\varphi_k,t-\tau)} \tag{S12}$$

S5 Original Visual Collective Motion Model (Castro *et al.*, 2024) [1]

Our model considers a system of N self-propelled particles confined to a plane (Fig. 1a-b). Each particle is represented as an object with a radius a, moving with a constant speed U. The position of each particle is denoted by $\mathbf{x}_i$, and its orientation is described by the unit vector $\mathbf{e}_i = [\cos\theta_i, \sin\theta_i]$. The collective motion model interacts with the particles by adjusting their angular velocity at each time step.

$$\dot{\mathbf{x}}_i = U\mathbf{e}_i, \tag{S13}$$

$$\dot{\theta}_i = k_{\odot}\,\omega_{\odot} + k_{\parallel}\,\omega_{\parallel} + k_{\eta}\,\eta, \tag{S14}$$

We represent the visual surroundings of the particles as retinal objects on a one-dimensional (1-D) binary panorama (Fig. 1c). This binary panorama is centered on the focal particle's heading, marking the presence of objects along the line of sight and generating binary retinal objects (Fig. 1d). Since all particles are perceived with the same magnitude, multiple particles may aggregate into a single, larger retinal object. Consequently, there are typically more i-particles than k-retinal objects. A apparent distance ($\mathcal{R}_i(\phi)$) is computed from this binary panorama (Fig. 1e), which is then used to calculate the attraction component of the angular velocity, $\omega_{\odot}$ (Eq. S15).

The optic flow perceived by each particle (illustrated by the black arrows in Fig. 1c) can be decomposed into two components: a radial component representing the translational optic flow (the average of the function shown in Fig. 1f), and an azimuthal component representing the divergence or convergence of the optic flow (Fig. 1g). The apparent velocity vector for each retinal object ($V_{i,k}$) can be determined using these two components of optic flow. This apparent velocity is employed to compute the alignment component of the angular velocity, $\omega_{\parallel}$ (Eq. S16). Finally, the individuality of each particle is incorporated into the angular velocity as a standard Wiener process (η).

$$\omega_{\odot} = \left\langle \int_{-\pi}^{\pi} \mathcal{R}_i^2(\phi) b_{\epsilon}(\phi)\sin\phi\, d\phi \right\rangle, \tag{S15}$$

$$\omega_{\parallel} = \left\langle \int_{-\pi}^{\pi} \frac{\mathbf{e}_i\times\mathbf{V}_{ik}}{U\mathcal{R}_i(\phi)} b_{\epsilon}(\phi)\, d\phi \right\rangle, \tag{S16}$$

$$\frac{\mathbf{e}_i\times\mathbf{V}_{ik}}{U\mathcal{R}_i(\phi)} = \frac{-\mathcal{D}_i(\phi)\sin\phi + \overline{\mathcal{O}}_i(\phi)\cos\phi}{U}. \tag{S17}$$

$b_{\epsilon}(\phi)$ is a parametric representation of an optic blind spot. The operand $\langle\cdot\rangle$ denotes a normalization that ensures the resulting function is of order $O(1)$. Conceptually, it functions like a min-max normalization on the amount of rotation for a single time step.

This model is made dimensionless by using a as the spatial unit and a/U as the temporal unit. The use of these spatial and temporal units creates a parameter space with the weights of each component of the angular velocity $(k_\odot, k_\parallel, k_\eta)$. Some Results can be seen on Fig. S6.

S6 Observed phases

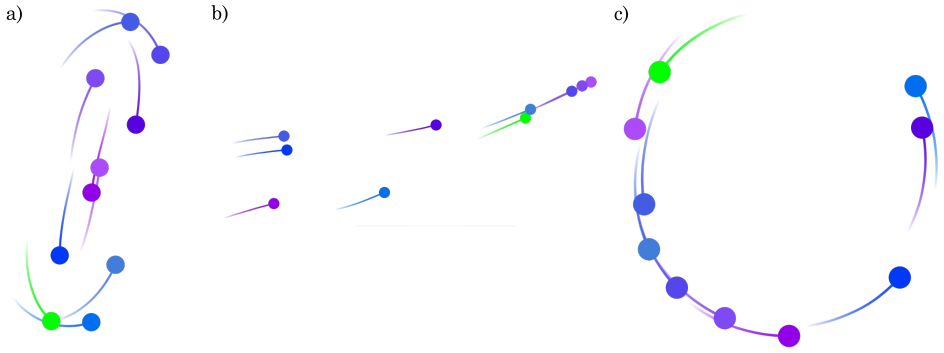

Fig. S6. Illustration of the phases observed for the raw original model [1]. We observe three phases when varying the other two parameters: (a) Swarming ($k_\odot = 0.15$, $k_\parallel = 0$) (see Svideo 13); (b) Schooling ($k_\odot = 0.07$, $k_\parallel = 0.3$) (see Svideo 14); (c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.05$) (see Svideo 15).

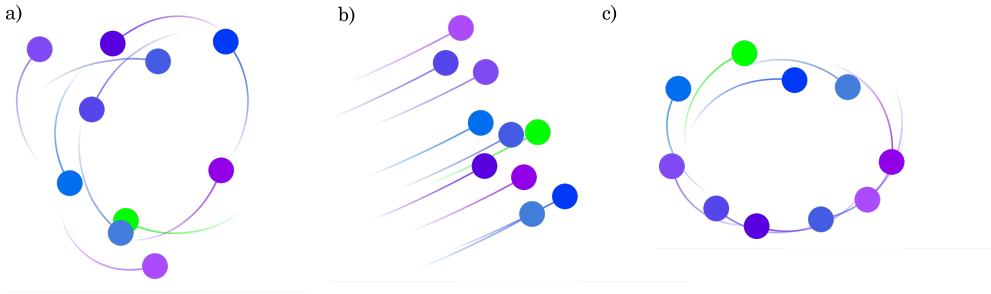

Fig. S7. Illustration of the phases observed for the visual model supplemented only with the avoidance. We observe three phases when varying the other two parameters: (a) Swarming ($k_\odot = 0.07$, $k_\parallel = 0$) (see Svideo 16); (b) Schooling ($k_\odot = 0.15$, $k_\parallel = 0.3$) (see Svideo 17); (c) Milling ($k_\odot = 0.07$, $k_\parallel = 0.05$) (see Svideo 18).

References

[1] Castro D, Ruffier F and Eloy C 2024 *Phys. Rev. Res.* **6**(2) 023016 URL `https://link.aps.org/doi/10.1103/PhysRevResearch.6.023016`

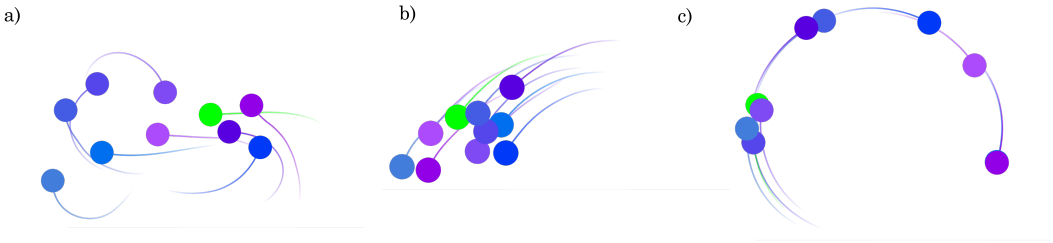

Fig. S8. Illustration of the phases observed for the visual model supplemented only with the avoidance and the visual anchor. We observe three phases when varying the other two parameters: (a) Swarming ($k_{\odot} = 0.3$, $k_{\parallel} = 0$) (see Svideo 19); (b) Schooling ($k_{\odot} = 0$, $k_{\parallel} = 0.3$) (see Svideo 20); (c) Milling ($k_{\odot} = 0.07$, $k_{\parallel} = 0.3$) (see Svideo 21).